

FVPP: Teorijske osnove formalne verifikacije provjerom modela

Propozicijska logika

Predikatna logika

Pripremio: izv. prof. dr. sc. Alan Jović

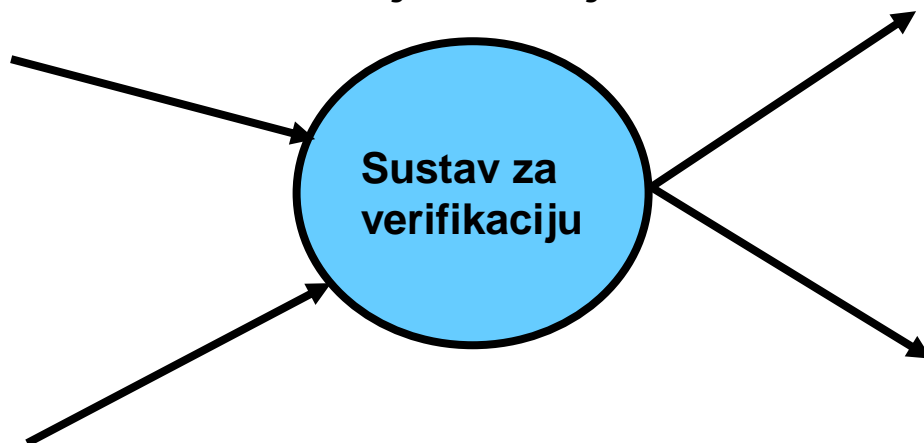
Ak. god. 2022./2023.



Provjera modela (engl. *model checking*)

I = Implementacija (model sustava koji se verificira). Izraženo povezanim strojevima s konačnim brojem stanja (FSM).

DA = model sustava logički zadovoljava specifikaciju



S = Specifikacija (željeno ponašanje). Izraženo najčešće u vremenskoj logici.

NE (+ ispis traga (engl. *trace*) pogrešnog izvođenja programa)

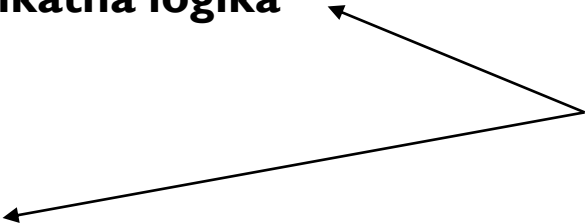
Simbolički opisujemo:

$$I \models S$$

Logika u provjeri modela

- Provjerom modela nastoji se dokazati **logička zadovoljivost** (“model implementacije zadovoljava specifikaciju”) – za to su potrebna znanja iz sljedećih područja:
 - **Formalna (matematička) logika**
 - **Modeliranje implementacije** strojevima s konačnim brojem stanja (u kontekstu formalne verifikacije model je tzv. Kripkeova struktura)
 - **Izražavanje specifikacije** (željenog ponašanja) vremenskom logikom kao proširenjem klasične matematičke logike

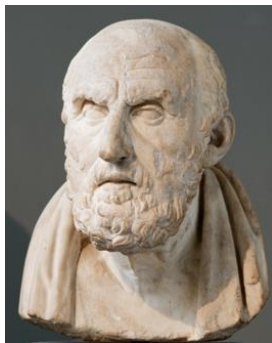
Formalna (matematička) logika

- Formalne logike su formalni jezici koji predstavljaju informaciju na način da se mogu automatizirano izvoditi "zaključci".
 - **Sintaksa** definira strukturu rečenice u jeziku.
 - **Semantika** definira značenje rečenica (definira istinitost rečenice u svijetu u kojem ju promatramo).
 - Postoji mnogo logika:
 - **Propozicijska i predikatna logika**
 - Logike višega reda
 - Modalne logike
 - Epistemička logika
 - **Vremenska logika**
 - ...
 - Opisna logika
 - Nemonotona logika
 - ...
- u ovom kolegiju
- 

Propozicijska logika (logika izjava, sudova, iskaza, tvrdnji)

Engl. *propositional logic, propositional calculus*

Hrizip iz Solija



Propozicijska logika - sintaksa

- Propozicijska logika preslikava deklarativne rečenice (koje mogu biti istinite ili lažne) u sustav simbola. Naprimjer: “Sokrat je mudar.” preslikava se u simbol P.
- Sustav propozicijske logike sastoji se od:
- PS: P, Q, ... PS je prebrojiv skup atoma, simboličkih varijabli, simbola
- Logički operatori (vezice):

\neg	(ne, not, \sim)	negacija
\wedge	(i, and, &)	konjunkcija
\vee	(ili, or,)	disjunkcija
\Rightarrow	(ako, if, \supset , \rightarrow)	implikacija
\Leftrightarrow	(akko, iff, \equiv , \leftrightarrow)	ekvivalencija

- Rezervirani simboli:

F	(false, \emptyset , 0, \perp)	konstanta (neistinitost)
T	(true, 1)	konstanta (istinitost)
(), .		znakovi zagrada, zareza i točke

- **Definicija** (rekurzivno) ispravno formiran (wff) složeni iskaz, ili formula:

1. Svaki atom je formula.

2. Ako su P i Q formule, onda su formule: $(\neg P)$, $(\neg Q)$, $(P \wedge Q)$, $(P \vee Q)$, $(P \Rightarrow Q)$, $(P \Leftrightarrow Q)$.

Propozicijska logika - semantika

- Pridruživanje obilježja istinitosti (T, F) atomičkim simbolima = **Interpretacija (I)**

- **I: PS \rightarrow BOOL**

gdje je $\text{BOOL} = \{ T, F \}$, tj. funkcija s kodomenom T ili F (istinito ili lažno).

Semantika dvaju složenih atomičkih simbola

- prikazuje se istinitosnom tablicom.
- 2 simbola = $2^2 = 4$ interpretacije, $2^4 = 16$ istinitosnih tablica = funkcija
- Neke važnije tablice istinitosti za povezivanje dvaju simbola:

	P Q		implikacija ($P \Rightarrow Q$)	ekvivalencija ($P \Leftrightarrow Q$)	kontradikcija (\neg), \perp	tautologija T
$I_1 :$	T	T	T	T	F	T
$I_2 :$	T	F	F	F	F	T
$I_3 :$	F	T	T	F	F	T
$I_4 :$	F	F	T	T	F	T

Semantička pravila – izračunavanje istinitosti formule

- Neka su: P_1, P_2 istinite, Q_1, Q_2 neistinite, A bilo koja formula (istinita ili ne). $=$ interpretacija

Istinite su formule:

$\neg Q_1$
 $(P_1 \wedge P_2)$
 $(P_1 \vee A)$
 $(A \vee P_1)$
 $(A \Rightarrow P_1)$
 $(Q_1 \Rightarrow A)$
 $(P_1 \Leftrightarrow P_2)$
 $(Q_1 \Leftrightarrow Q_2)$
 $(A \vee \neg A)$ - tautologija

Neistinite su formule:

$\neg P_1$
 $(Q_1 \wedge A)$
 $(A \wedge Q_1)$
 $(Q_1 \vee Q_2)$
 $(P_1 \Rightarrow Q_1)$
 $(P_1 \Leftrightarrow Q_1)$
 $(Q_1 \Leftrightarrow P_1)$
 $(A \wedge \neg A)$ - kontradikcija
 $()$ - prazna formula

- Primjer izračunavanja istinitosti složene formule s 3 propozicijska simbola P, Q, R :
 $(Q \vee (((\neg Q) \wedge P) \Rightarrow R))$
- Interpretacija** (3 simbola povlači 2^3 mogućih interpretacija)
- Neka je jedna interpretacija I : $P=T, Q=F, R=F$,
- Izračunavanje (**evaluacija**) temeljem osnovnih tablica istinitosti i računajući "iznutra prema van" daje ovoj formuli neistinitu vrijednost.
- Semantika uključuje interpretaciju i evaluaciju.**

Pravila ekvivalencije

- Definicija: Dvije formule su semantički **ekvivalentne** ili **jednake** ako imaju jednaku (istu) istinitosnu vrijednost **za svaku interpretaciju**.
- Provjera koincidencije istinitosnih tablica **nije u općem slučaju dovoljna** jer formule ne moraju sadržavati iste simbole. Vidjeti za opći slučaj sl. 20.

- | | |
|--|---------------------------------|
| • $(A \wedge \neg A) = ()$ | kontradikcija |
| • $(\neg(\neg A)) = A$ | dvostruka negacija |
| • $(A \wedge A) = A$ | jednaka važnost (idempotencija) |
| • $(A \vee A) = A$ | jednaka važnost |
| • $(A \vee B) = (B \vee A)$ | komutativnost |
| • $(A \wedge B) = (B \wedge A)$ | komutativnost |
| • $((A \vee B) \vee C) = (A \vee (B \vee C))$ | asocijativnost |
| • $((A \wedge B) \wedge C) = (A \wedge (B \wedge C))$ | asocijativnost |
| • $(A \wedge (B \vee C)) = ((A \wedge B) \vee (A \wedge C))$ | distributivnost |
| • $(A \vee (B \wedge C)) = ((A \vee B) \wedge (A \vee C))$ | distributivnost |
| • $(\neg(A \vee B)) = ((\neg A) \wedge (\neg B))$ | De Morganov zakon |
| • $(\neg(A \wedge B)) = ((\neg A) \vee (\neg B))$ | De Morganov zakon |
| • $(A \Rightarrow B) = ((\neg A) \vee B)$ | eliminacija uvjeta |
| • $(A \Leftrightarrow B) = ((A \Rightarrow B) \wedge (B \Rightarrow A))$ | eliminacija dvostrukog uvjeta |
| • $(A \Rightarrow B) = ((\neg B) \Rightarrow (\neg A))$ | transpozicija |

Formalan sustav

- Matematička ili formalna logika daje sustav zaključivanja u kojem je "logički izveden" zaključak barem tako dobar kao polazne pretpostavke.
- Temelj formalne logike: **formalan sustav**
- Definiramo formalan sustav kao dvojku: $\{\Gamma, L\}$ u odabranoj logici gdje je
 - Γ – skup ispravno definiranih (formiranih) formula (wff)
 - L – konačan skup pravila zaključivanja

Temeljna pravila zaključivanja L

- Generiraju dodatne istinite formule (mehanički) iz početnih formula – **aksioma** formalnog sustava bez razumijevanja konteksta (značenja).
- Pogodna za strojnu primjenu.
- Ako $P=T$, $Q=T$, generiraj $(P \wedge Q) = T$ (uvođenje $\wedge = \wedge_i$)
- Ako $P=T$, $(P \Rightarrow Q)=T$, generiraj $Q = T$ ("modus ponens")
- Ako $\neg Q=T$, $(P \Rightarrow Q)=T$, generiraj $\neg P$ ("modus tolens")
- Ako $(P \wedge Q)=T$, generiraj $(Q \wedge P) = T$ (komutativnost \wedge)
- Ako $(P \wedge Q)=T$, generiraj $P = T, Q = T$ (eliminacija $\wedge = \wedge_e$)
- Ako $P=T$ (odnosno $Q=T$), generiraj $(P \vee Q)$ (uvođenje $\vee = \vee_i$)
- Ako $[\neg(\neg P)]=T$, generiraj $P = T$ (eliminacija $\neg = \neg_e$)
- Kroz primjere za vježbu pokazat će se još i pravilo “eliminacija \vee ” kao i pravilo “uvođenje \Rightarrow ”, vidjeti i zadnji slajd

Terminologija formalnih sustava

Dedukcija (engl. deduction)

- Sekvencija formula $\{\omega_1, \omega_2, \dots, \omega_n\}$ ili pojedina formula ω_i je **dedukcija (dokaz)** iz skupa formula Γ ako se već nalazi u skupu formula Γ (tada se naziva aksiom) ili se može izvesti iz Γ korištenjem pravila zaključivanja **L**.

$\Gamma \vdash_L \{\omega_1, \omega_2, \dots, \omega_n\}$ sekvencija formula je dedukcija od Γ

$\Gamma \vdash_L \omega_i$ formula ω_i je dedukcija od Γ

Primjeri

- Neka skup Γ sadrži dvije istinite formule: $\Gamma = \{P, (P \Rightarrow Q)\}$

Korištenjem pravila “**Modus ponens**” (iz skupa dopustivih pravila **L**), izvodimo da je istinita nova formula Q , te je ta formula Q dedukcija (dokaz) skupa Γ .

- Neka skup Γ sadrži tri istinite formule: $\Gamma = \{P, Q, (Q \wedge R)\}$

Formule P , Q , i $(Q \wedge R)$ su aksiomi, a ujedno i dedukcije, dok je formula R dedukcija, jer to daje pravilo eliminacije \wedge .

Terminologija formalnih sustava

Teorem (engl. *theorem*)

- Formula ω_i je **teorem** ako se može izvesti korištenjem pravila zaključivanja **L** iz praznog skupa formula (bez premisa ili aksioma)

$\vdash_L \omega_i$ formula ω_i je teorem

Primjeri teorema

$\vdash_L ((P \Rightarrow Q) \wedge (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R)$

$\vdash_L Q \Rightarrow (P \Rightarrow (P \Rightarrow (Q \Rightarrow P)))$

Terminologija formalnih sustava

Konzistentnost (engl. *consistency*)

- Skup formula Γ je **konzistentan** ako i samo ako ne sadrži formule na temelju kojih bi ω_i i $\neg\omega_i$ (istovremeno) bile dedukcije.

Primjeri

- $\Gamma = \{ P, (P \Rightarrow Q) \}$ je konzistentan skup.
- $\Gamma = \{ P, \neg P, (P \Rightarrow Q) \}$ je **nekonzistentan** ili **kontradiktoran** jer su P i $\neg P$ istovremeno dedukcije (kontradiktorni aksiomi se nalaze već u samom skupu Γ).
- $\Gamma = \{ P, \neg Q, (P \Rightarrow Q) \}$ je nekonzistentan jer sadrži $\neg Q$, a pravilom “Modus ponens” može se izvesti Q , dakle $\neg Q$ i Q bi istovremeno bile dedukcije.

Terminologija formalnih sustava

Odredivost (odlučljivost, engl. *decidability*)

- Neka se u formalnom sustavu $\{\Gamma, L\}$ izvodi neka formula ω_i (tražimo odgovor je li ω_i dedukcija).
- Formalan sustav je **odrediv** ili **odlučljiv** (engl. *decidable*), ako i samo ako postoji postupak, procedura ili **algoritam** koji će u konačnom vremenu odrediti ili ne dedukciju ω_i (dati u konačnom vremenu odgovor da je ω_i dedukcija ili da ω_i nije dedukcija).
- Formalan sustav $\{\Gamma, L\}$ je **poluodrediv** ili **poluodlučljiv** (engl. *semidecidable*), ako i samo postoji algoritam koji će u konačnom vremenu odrediti dedukciju ako ona postoji. Algoritam završava u konačnom vremenu s odgovorom "da" (za dedukciju ω_i), ali ne mora završiti u konačnom vremenu s odgovorom "ne" (ako ω_i nije dedukcija). Moguća je i alternativa (završava za "ne", a ne mora završiti za "da")
- Formalan sustav je **neodrediv** ili **neodlučljiv** (engl. *undecidable*) ako nije ni odrediv ni poluodrediv.

Terminologija formalnih sustava

Semantika u formalnom sustavu povezana je s:

- **interpretacijom** - pridruživanjem istinitosti atomima i
- **evaluacijom** - izračunavanjem istinitosti složene formule.

Model (engl. *model*)

- Neka interpretacija je **model** formalnog sustava $\{\Gamma, \mathbf{L}\}$ ako evaluira **sve** njegove formule u istinito

Primjer:

I: $\{P=T, Q=F, R=F\}$ formule $(Q \vee (((\neg Q) \wedge P) \Rightarrow R))$

nije model jer ta interpretacija formuli daje neistinitu vrijednost.

Logička zadovoljivost (engl. *logical satisfiability*)

- Skup formula je logički **zadovoljiv** ako ima (barem jedan) model.
- Vrijedi i za pojedinačne formule.
- Sukladno ranijoj definiciji, **logički nezadovoljiv** (**nekonzistentan**, **kontradiktoran**) skup formula nema nijedan model.

Terminologija formalnih sustava

Logička posljedica (engl. *logical consequence*)

- Formula ω je **logička posljedica** skupa formula Γ , ako je svaki model od Γ ujedno i model od ω
- Kažemo i da skup formula Γ **povlači** (engl. *entails*) formulu ω
- Oznaka logičke posljedice:
 $\Gamma \models \omega$

Valjanost

- Formula je **valjana** (engl. *valid*) ili **tautologija** (engl. *tautology*) ako je istinita **za svaku** interpretaciju i evaluaciju.
- Oznaka valjane formule:
 $\models \omega$ (svaka interpretacija je model formule ω)

Terminologija formalnih sustava

- Logička posljedica izrečena na drugi način: Ako svaka interpretacija koja lijevoj strani od znaka \models daje istinitost ujedno daje i desnoj strani istinitost, tada je desna strana logička posljedica lijeve.

Primjeri logičkih posljedica

- 1. $(P \wedge Q) \models P$

lijeva strana = T samo za $(P=T, Q=T)$, samo jedan model, a to daje i desnoj strani =T, dakle gornji izraz vrijedi (P je logička posljedica $(P \wedge Q)$).

- 2. $(P \vee Q) \models P$

lijeva strana je istinita za $(P=F, Q=T; P=T, Q=F; P=T, Q=T)$, ali desna za interpretaciju $(P=F, Q=T)$ nije istinita, te P **nije logička posljedica** $(P \vee Q)$.

- 3. $\{\neg Q, (P \vee Q)\} \models P$ (zarez predstavlja konjunkciju \wedge)

skup Γ na lijevoj strani je istinit samo za $Q=F, P=T$, a to daje istinitost i desnoj strani, te je P logička posljedica navedenog skupa Γ .

- 4. $P \models (Q \vee \neg Q)$

također vrijedi, jer za svaku interpretaciju za koju je lijeva strana istinita ($P=T$) i desna strana je istinita (desna strana je doduše uvijek istinita).

Terminologija formalnih sustava

- Skup formula Γ naziva se još **baza znanja** (engl. *knowledge base*, **KB**) formalnog sustava

Primjer: $\Gamma = \{(A \vee C) \wedge (B \vee \neg C)\}$ = baza znanja = dvije konjunkcijom povezane formule (umjesto \wedge može se koristiti zarez).

Neka je: $\alpha = (A \vee B)$

Pitanje: **KB** $\models \alpha$?

DA!

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
<i>False</i>	<i>False</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>False</i>
<i>False</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>False</i>	<i>False</i>
<i>False</i>	<i>True</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>False</i>	<i>True</i>
<i>False</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<u><i>True</i></u>	<u><i>True</i></u>
<i>True</i>	<i>False</i>	<i>False</i>	<i>True</i>	<i>True</i>	<u><i>True</i></u>	<u><i>True</i></u>
<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>	<i>False</i>	<i>False</i>	<i>True</i>
<i>True</i>	<i>True</i>	<i>False</i>	<i>True</i>	<i>True</i>	<u><i>True</i></u>	<u><i>True</i></u>
<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<i>True</i>	<u><i>True</i></u>	<u><i>True</i></u>

Terminologija formalnih sustava

Semantička ekvivalentnost: stroža definicija ekvivalencije dviju formula preko pojma logičke posljedice (\models)

- Dvije formule α i β su **semantički ekvivalentne** (oznake $(\alpha \Leftrightarrow \beta)$ ili $(\alpha \equiv \beta)$) ako i samo ako vrijede (istinite su) logičke posljedice: **$(\alpha \models \beta)$ i $(\beta \models \alpha)$** .
- Ranija tablica pravila ekvivalencije daje:

$$(\alpha \Leftrightarrow \beta) = (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha).$$

- Ako su α i β ekvivalentne, formula $((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$ mora uvijek biti istinita:

$$\models ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$$

- Semantička ekvivalencija je na taj način identična dokazljivoj ekvivalenciji: ako želimo dokazati ekvivalentnost dviju formula, dokažemo da je formula **$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$ tautologija** ili da je njena negacija nezadovoljiva.

Terminologija formalnih sustava

Ispravnost (engl. *soundness*) i kompletnost (engl. *completeness*)

- Formalan sustav $\{\Gamma, \mathbf{L}\}$ je **ispravan** (engl. *sound*) ako je svaka (pravilima dokazana) dedukcija ujedno i logička posljedica skupa formula Γ , tj.:

$$\Gamma \vdash_{\mathbf{L}} \omega_i \quad \Rightarrow \quad \Gamma \models \omega_i$$

Neformalno, ispravnost osigurava da je svaka činjenica koja se dokaže istinita.

- Formalan sustav $\{\Gamma, \mathbf{L}\}$ je **kompletan** (engl. *complete*) ako je svaku logičku posljedicu skupa Γ moguće dokazati pravilima \mathbf{L} , tj.:

$$\Gamma \models \omega_i \quad \Rightarrow \quad \Gamma \vdash_{\mathbf{L}} \omega_i$$

Neformalno, kompletnost osigurava da se mogu dokazati **sve** istinite činjenice.

- Primijetiti: u kompletnom sustavu vrijedi: $\models \omega_i \Rightarrow \vdash_{\mathbf{L}} \omega_i$
- U ispravnom i kompletnom formalnom sustavu $\{\Gamma, \mathbf{L}\}$ vrijedi:
- $\Gamma \models \omega_i \quad = \quad \Gamma \vdash_{\mathbf{L}} \omega_i$ (logička posljedica je ujedno dedukcija i obratno)

Terminologija formalnih sustava

Većina interesantnih formalnih logičkih sustava je nekompletna, a vrlo malo ih je odredivo.

Propozicijska logika je:

- Ispravna, kompletna i odrediva (npr. preslikavanjem u tablicu istinitosti), jer operira s konačnim skupom simbola.

Predikatna logika prvoga reda je:

- Poluodrediva (ako dedukcija postoji, dokazat će se, a ako ne postoji može se ali i ne mora dokazati).
- Odredivi su samo neki podskupovi logike prvog reda.
- "Čista" (npr. bez aritmetike) predikatna logika je ispravna i kompletna (Gödel).

Predikatne logike višega reda nisu kompletne pod pretpostavkom pune semantike.

Normalni oblici propozicijskih formula

- Svaka formula propozicijske logike može se preslikati (ekvivalentna je) formuli u disjunkcijskom normalnom obliku (**DNF**) :

$$(k1_1 \wedge \dots \wedge k1_n) \vee (k2_1 \wedge \dots \wedge k2_m) \vee \dots \vee (kp_1 \wedge \dots \wedge kp_r)$$

- Svaka propozicijska formula može se preslikati (ekvivalentna je) formuli u konjunksijskom normalnom obliku (**CNF**) :

$$(k1_1 \vee \dots \vee k1_n) \wedge (k2_1 \vee \dots \vee k2_m) \wedge \dots \wedge (kp_1 \vee \dots \vee kp_r)$$

gdje su:

- k_i = literal (negirani ili nenegirani atomički simbol - atom)
- **klauzula = disjunkcija literala.** Npr.: $(k2_1 \vee \dots \vee k2_m)$
- Preslikavanje CNF u DNF i obrnuto je računalno vrlo skupo (vremenski i prostorno). Spada u razred NP-teških (engl. *NP-hard*) problema.

Pretvorba u normalni oblik CNF

Svaka formula u propozicijskoj logici može se pretvoriti u konjunkciju klauzula (CNF):

$$\text{Npr: } \neg(P \Rightarrow Q) \vee (R \Rightarrow P)$$

Algoritam:

1. Eliminiraj ekvivalencije i implikacije uporabom ekvivalentnog " \vee " oblika:

$$\neg(\neg P \vee Q) \vee (\neg R \vee P)$$

2. Reduciraj doseg negacije (pomak u desno) uporabom DeMorganovih pravila, te eliminiraj dvostruke negacije:

$$(P \wedge \neg Q) \vee (\neg R \vee P)$$

3. Pretvori u CNF asocijativnim i distribucijskim pravilima:

$$(P \vee \neg R \vee P) \wedge (\neg Q \vee \neg R \vee P),$$

te dalje:

$$(P \vee \neg R) \wedge (\neg Q \vee \neg R \vee P) \quad = \text{CNF oblik}$$

- Napomena: pretvorba u CNF-oblik ne dovodi nužno do minimalnog CNF-oblika, minimizacija se naknadno provodi prema potrebi.**

SAT-problem – temeljni NP problem

- Traži se model skupa formula Γ (interpretaciju koja evaluira sve formule u skupu Γ u istinito. To je ekvivalentno traženju modela **jedne** složene formule koja se sastoji iz **konjunkcije svih formula u Γ** .
- CNF-oblik skupa formula:
$$(k1_1 \vee \dots \vee k1_p) \wedge (k2_1 \vee \dots \vee k2_r) \wedge \dots \wedge (kp_1 \vee \dots \vee kp_s)$$
- Iscrpna procedura rješavanja **CNF SAT**-problema sistematski pridjeljuje istinitosne vrijednosti atomičkim propozicijskim simbolima. **Za n atoma 2^n pridruživanja.** Ekspnencijalna složenost!
- CNF 2SAT (do 2 literala u klauzuli) - polinomna složenost
- **CNF 3SAT (3 literala u klauzuli) - NP-kompletno**, ekspnencijalna složenost
- **Zadovoljivost formule u CNF-obliku s 3 i više literala je NP-kompletno.**
- Mnogi stohastički algoritmi troše ekspnencijalno vrijeme u najgorem slučaju, ali polinomno u srednjem (očekivanom). Ti se algoritmi zovu SAT-rješavači (engl. *SAT solver*).
- Najpoznatiji suvremeni SAT-solveri: zChaff, miniSAT, SatZ

Teorem dedukcije

- **Teorem:** Formula ψ je **logička posljedica** formule φ , tj. $\varphi \models \psi$, ako i samo ako je formula $(\varphi \Rightarrow \psi)$ tautologija (valjana).
- **Dokaz:** Ako je $(\varphi \Rightarrow \psi)$ tautologija (uvijek istinita), onda iz tablice za implikaciju proizlazi da kada je φ istinit da tada i ψ mora biti istinit (jer bi alternativa vodila na neistinu). To je upravo definicija logičke posljedice.

φ	ψ	$(\varphi \Rightarrow \psi)$
F	F	T
F	T	T
T	F	F
T	T	T

- **Korolar (metoda opovrgavanja):** Budući da $(\varphi \Rightarrow \psi)$ mora biti tautologija, to njena negacija $\neg(\varphi \Rightarrow \psi) = \neg(\neg\varphi \vee \psi) = (\varphi \wedge \neg\psi)$ mora biti nezadovoljiva. Dakle:

$\varphi \models \psi$ akko je $(\varphi \wedge \neg\psi)$ nezadovoljiva

Primjer rasuđivanja opovrgavanjem

Neka istinite formule predstavljaju skup Γ :

1. P
2. $(P \Rightarrow Q)$
3. $(Q \Rightarrow S)$

U CNF obliku: $\Gamma = [(P) \wedge (\neg P \vee Q) \wedge (\neg Q \vee S)]$

- Pitamo se: je li S logička posljedica skupa Γ : $\Gamma \models S$?
- **Teorem dedukcije i korolar: S je logička posljedica Γ ako je $(\Gamma \wedge \neg S)$ nezadovoljiva.**
- Dakle, skupu Γ dodajemo negaciju formule koju želimo dokazati ($\neg S$):
 $[(P) \wedge (\neg P \vee Q) \wedge (\neg Q \vee S) \wedge (\neg S)]$

Sad možemo iskoristiti npr. SAT-rješavač da pokušamo naći barem jedan model (zadovoljivost). Ako SAT-rješavač pokaže da formulu **nije** moguće zadovoljiti (opovrgnuo ju je – nema modela), zaključujemo:

S je doista logička posljedica skupa Γ .

Zadaci

1. Dokažite istinitost sljedećih logičkih zaključaka:

a) $(P \wedge Q) \wedge R, S \wedge T \vdash_L Q \wedge S$

b) $P \wedge Q \Rightarrow R \vdash_L P \Rightarrow (Q \Rightarrow R)$

c) $P \Rightarrow Q \vdash_L \neg P \vee Q$

2. Pretvorite propozicijsku formulu u CNF-oblik

$$(P \Rightarrow (Q \Rightarrow R)) \Leftrightarrow (P \Rightarrow (R \Rightarrow Q))$$

3. Koristeći teorem o dedukciji, ekvivalencije koje vrijede u propozicijskoj logici i pravila zaključivanja, pokažite da je formula S logička posljedica skupa formula:

1. P

2. $(P \Rightarrow Q)$

3. $(Q \Rightarrow S)$

4. Koristeći pojam logičke posljedice, pokažite ili opovrgnite semantičku ekvivalentnost formula

$\varphi_1: P \Rightarrow (Q \vee R)$ i

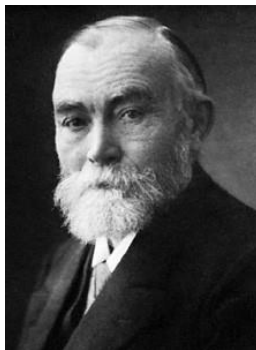
$\varphi_2: (\neg Q \wedge \neg R) \Rightarrow P$

Predikatna logika

(Logika predikata prvoga reda)

Engl. *predicate logic, predicate calculus, first order predicate logic* – FOPL

Gottlob Frege



Kurt Gödel



Alfred Tarski



Predikatna logika – sintaksa

1. P: Svi ljudi su smrtni.

2. Q: Sokrat je čovjek.

3. R: Sokrat je smrtan.

- U propozicijskoj logici nikako se iz 1:P i 2:Q ne može zaključiti 3:R.
FOPL uvodi objekte, relacije, obilježja, funkcije (**za pobliži opis izjave**).
Povećana je izražajna moć formalne logike.

Sintaksa:

Atomički predikat:

(pred_simb t1 t2 ... tn)	– infiks notacija (LISP)	} oba načina pisanja OK, ali ne miješati
pred_simb(t1 t2 ... tn)	– prefiks notacija (Prolog)	

- **pred_simbol:** osnovno obilježje u rečenici (predikat)
- **t_i = članovi:** objekti ili odnosi u rečenici

Predikatna logika – sintaksa

Članovi (t_i) :

- Konstante: objekti u nekom svijetu (blok I, sokrat, ...).
- Rezervirane konstante: T, F.
- Varijable: razred objekata ili obilježja; mogu poprimiti vrijednosti iz svoje domene; (Npr.: X, Y, ...).
- Funkcije: veza između objekata - (fun_simb t_1 t_2 ... t_n). Npr.: (cos X), (otac_od abel kain)

Formalna def. člana:

- 1. Konstanta je član.
- 2. Varijabla je član.
- 3. Ako je fun_simb funkcijski simbol sa n-argumenata, a t_1, t_2, \dots, t_n su članovi, tada je (fun_simb t_1 t_2 ... t_n) član.

Logički operatori (vezice): $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Kvantifikacijski simboli (uz varijable, pobliže određuju istinitost rečenice):

- \exists (postoji, za_neki, exist) - egzistencijski ili partikularni kvantifikator (barem jedan).
- \forall (za_svaki, svi, for_all) - univerzalni kvantifikator (svi), ima središnju ulogu u izražavanju generalizacije.

Predikatna logika – sintaksa

Ispravno definiran složeni predikat ili formula (wff):

1. svaki atomički predikat je formula.
 2. ako je S_i formula, tada su formule:
 $(\neg S), (S_1 \wedge S_2), (S_1 \vee S_2), (S_1 \Rightarrow S_2), (S_1 \Leftrightarrow S_2).$
 3. ako je X varijabla, a S formula, tada su formule: $\exists X S(X), \forall X S(X).$
oznaka $S(X)$ = formula S u kojoj postoji varijabla X
- Negirani ili nenegirani atomički predikat naziva se **literal**.

Primjer ispravno definirane složene formule u infiks notaciji:

$$(\forall X \forall Y (((\text{otac } XY) \vee (\text{majka } XY)) \Rightarrow (\text{roditelj } XY)))$$

Predikatna logika – semantika

Skup wff uvijek se odnosi na neku domenu razmatranja D .

Interpretacija (I) je proces preslikavanja elemenata iz domene D svakoj pojedinoj konstanti, varijabli, i funkciji, te atomičkom predikatu, tako da:

- Simbolu T uvijek je pridružena istinita vrijednost.
- Simbolu F uvijek je pridružena neistinita vrijednost.
- Svakoj konstanti pridruži se jedan element iz D .
- Svakom funkcijskom simbolu pridruži se jedan element iz D .
- Svakoj varijabli se pridruži neprazan podskup iz D (dozvoljene supstitucije).
- Svaka funkcija f , sa m argumenata, definira **interpretacijom i evaluacijom** preslikavanje iz D^m u D , tj. $f: D^m \rightarrow D$ (funkcija se evaluira u jedan element iz D).
- Svaki predikat P , s brojem članova n , definira interpretacijom i evaluacijom svojih članova preslikavanje iz D^n u $\{T, F\}$, tj. $P: D^n \rightarrow \{T, F\}$ (predikat se za određene elemente domene preslika u istinu ili laž).
- Vrijednosti wff formula složenih logičkim operatorima dane su odgovarajućim istinitosnim tablicama.
- Vrijednost $\forall X P(X)$ je T , ako $P(X)$ je T , za sve vrijednosti X dane sa I , a F inače.
- Vrijednost $\exists X P(X)$ je T , ako $P(X)$ je T , barem za jednu vrijednost X danoj sa I , a F inače.

Predikatna logika – semantika

- Određivanje istinitosti wff svodi se na **interpretaciju + evaluaciju**
- Skup svih istinitih predikata nad domenom D naziva se **stanje svijeta** (*engl. state of the world*).

Primjeri pridruživanja istinitosti:

1. (prijatelj ivan ana)

predikat je T, ako u D postoji objekt Ana koja je prijatelj Ivanu.

2. Neka je domena od X skup prirodnih brojeva. Tada:

$\forall X$ (veci X 10)

atomički predikat je F

$\exists X$ (veci X 10)

atomički predikat je T

- \forall - u određivanju T potrebne sve supstitucije varijable (problem ako je domena beskonačna)
- \exists - u određivanju T potrebna jedna supstitucija za koju T (problem ako je domena beskonačna i predikat F)

Dopuna pravilima ekvivalencije

Simbol varijable i doseg kvantifikatora:

Neka su $P(X)$, $Q(X)$ wff s varijablom X , tada vrijedi:

$$\exists X P(X) = \exists Y P(Y)$$

$$\forall X P(X) = \forall Y P(Y)$$

- simbol varijable nije bitan, ali je bitan doseg, uvijek unutar jedne formule

Proširenje De Morganovih relacija:

Primjer: "*Ne vole svi ići zubaru.*" = "*Postoji netko tko ne voli ići zubaru.*"

$$\neg(\forall X P(X)) = \exists X (\neg P(X))$$

$$\neg(\exists X Q(X)) = \forall X (\neg Q(X))$$

- **Negacija mijenja kvantifikator!**

Dopuna pravilima ekvivalencije

Supstitucija

- Neka je **P(X)** wff s varijablom X.
- Neka je domena X: **D = {1, 2, 3}**
- Formula $\forall X P(X)$ je ekvivalentna $[P(1) \wedge P(2) \wedge P(3)]$
 $\forall X P(X) \equiv [P(1) \wedge P(2) \wedge P(3)]$
- $\forall X P(X)$ je istinita ako su istinite **sve** supstitucije iz domene.
- Formula $\exists X P(X)$ je ekvivalentna $[P(1) \vee P(2) \vee P(3)]$
 $\exists X P(X) \equiv [P(1) \vee P(2) \vee P(3)]$
- $\exists X P(X)$ je istinita ako je istinita **barem jedna** supstitucija iz domene.

Permutacija kvantifikatora

- Formule :
 $\forall X \exists Y P(X, Y) \neq \exists Y \forall X P(X, Y)$ - nisu ekvivalentne!
Npr. $\forall x \exists y \text{Voli}(x, y)$: svatko voli nekoga i $\exists y \forall x \text{Voli}(x, y)$: postoji netko koga svi vole

Ispravna uporaba univerzalnog kvantifikatora \forall

Primjer:

- Neka je okvir razmatranja (skup objekata): {Garfield, Feliks, računalo}
- Preslikaj u predikatnu logiku rečenicu: "Sve mačke su sisavci."
- **Za sve** objekte x u okviru razmatranja vrijedi: ako su mačke tada su sisavci.

$$\forall x [\text{mačka}(x) \Rightarrow \text{sisavac}(x)]$$

- Dokaz: Supstitucija svih objekata u formulu (konjunkcija formula jer \forall):

$$[\text{mačka}(\text{Garfield}) \Rightarrow \text{sisavac}(\text{Garfield})] \wedge [\text{mačka}(\text{Feliks}) \Rightarrow \text{sisavac}(\text{Feliks})] \wedge [\text{mačka}(\text{računalo}) \Rightarrow \text{sisavac}(\text{računalo})]$$

- prva $[T \Rightarrow T]$: T (vidi tablicu za \Rightarrow)
- druga $[T \Rightarrow T]$: T (vidi tablicu za \Rightarrow)
- treća $[F \Rightarrow F]$: T (vidi tablicu za \Rightarrow)

time je ukupna formula = T !!!

Ispravna uporaba univerzalnog kvantifikatora \forall

Primjer (nastavak):

- Da smo napisali: $\forall x [\text{mačka}(x) \wedge \text{sisavac}(x)]$

Doslovno: “svaki x je mačka i svaki x je sisavac”

- Supstitucija svih objekata u tom slučaju daje:
 $[\text{mačka}(\text{Garfield}) \wedge \text{sisavac}(\text{Garfield})] \wedge [\text{mačka}(\text{Feliks}) \wedge \text{sisavac}(\text{Feliks})] \wedge$
 $[\text{mačka}(\text{računalo}) \wedge \text{sisavac}(\text{računalo})]$
- $\text{mačka}(\text{računalo}) = F$ - **daje neistinitu cijelu formulu !!!**

Ispravna uporaba egzistencijskog kvantifikatora \exists

Primjer:

- Neka je okvir razmatranja (kao i prije): {Garfield, Feliks, računalo}
- Preslikaj u predikatnu logiku: "Garfield ima brata koji je mačka."
- Postoji **barem jedan** (neki) objekt i takav da su mu obilježja istinita.
 $\exists x [\text{brat}(x, \text{Garfield}) \wedge \text{mačka}(x)]$
- Dokaz supstitucijom svih objekata u formulu (disjunkcija formula jer \exists):

$[\text{brat}(\text{Garfield}, \text{Garfield}) \wedge \text{mačka}(\text{Garfield})] \vee$
 $[\text{brat}(\text{Feliks}, \text{Garfield}) \wedge \text{mačka}(\text{Feliks})] \vee$
 $[\text{brat}(\text{računalo}, \text{Garfield}) \wedge \text{mačka}(\text{računalo})]$

- Prva [] neistinita jer Garfield nije sam sebi brat, ali idemo dalje jer su [...] povezane disjunkcijom.
- Drugi red istinit, cijela formula je istinita (dalje ne moramo ispitivati).

Ispravna uporaba egzistencijskog kvantifikatora \exists

Primjer (nastavak):

- Ako bi preslikali: $\exists x [\text{brat}(x, \text{Garfield}) \Rightarrow \text{mačka}(x)]$
- To se drugačije može napisati kao: $\exists x [\neg \text{brat}(x, \text{Garfield}) \vee \text{mačka}(x)]$ – doslovno: “postoji takav x koji ili nije brat od Garfielda ili je mačka”
- Supstitucija svih objekata u disjunkciju formula daje:
 $[\text{brat}(\text{Garfield}, \text{Garfield}) \Rightarrow \text{mačka}(\text{Garfield})] \vee$
 $[\text{brat}(\text{Feliks}, \text{Garfield}) \Rightarrow \text{mačka}(\text{Feliks})] \vee$
 $[\text{brat}(\text{računalo}, \text{Garfield}) \Rightarrow \text{mačka}(\text{računalo})]$
- Implikacija je istinita ako je atomički izraz na lijevoj strani neistinit!
- Npr. ako je: $[\text{brat}(\text{računalo}, \text{Garfield}) \Rightarrow \text{mačka}(\text{računalo})]$ istinito, cijela je formula istinita !
- Egzistencijski kvantificirana implikacijska formula je istinita ako u okviru razmatranja postoji barem jedan objekt za koji je premisa implikacije **neistinita** (desna strana može biti T ili F).
- Takva rečenica **ne daje nikakvu potvrdnu informaciju**.
- **Zaključak:** $\forall \text{ ide uz } \Rightarrow$
 $\exists \text{ ide uz } \wedge$

Obilježja predikatne logike

- Zadovoljivost
- Model
- Logička posljedica
- Kontradiktornost
- Pravila zaključivanja

Sva navedena svojstva su jednaka kao i u propozicijskoj logici.

- **Predikatna logika višega reda (engl. *Higher-Order Logic*):**
 - Kvantifikacija na predikatnom (ili funkcijskom) simbolu.

Npr: $\forall (\text{Voli}) (\text{Voli ivo ana})$

Zadaci

I. Preslikajte sljedeće rečenice prirodnog jezika u formalizam predikatne logike prvoga reda (FOPL). Pritom definirajte sve potrebne predikate i konstante.

a) „Niti jedan student ne sluša sve predmete.”

b) „Svaki profesor je zaposlenik samo jednog fakulteta, a predaje na jednom ili više fakulteta.”

c) „Svatko voli nekog i nitko ne voli svakog.”

d) „Neki studenti koji slušaju predmet FMuOS također slušaju i predmet NOS.”

introduction

elimination

$$\wedge \quad \frac{\phi \quad \psi}{\phi \wedge \psi} \wedge i \quad \frac{\phi \wedge \psi}{\phi} \wedge e_1 \quad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

$$\vee \quad \frac{\phi}{\phi \vee \psi} \vee i_1 \quad \frac{\psi}{\phi \vee \psi} \vee i_2 \quad \frac{\phi \vee \psi \quad \boxed{\begin{smallmatrix} \phi \\ \vdots \\ \chi \end{smallmatrix}} \quad \boxed{\begin{smallmatrix} \psi \\ \vdots \\ \chi \end{smallmatrix}}}{\chi} \vee e$$

$$\rightarrow \quad \frac{\boxed{\begin{smallmatrix} \phi \\ \vdots \\ \psi \end{smallmatrix}}}{\phi \rightarrow \psi} \rightarrow i \quad \frac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow e$$

$$\neg \quad \frac{\boxed{\begin{smallmatrix} \phi \\ \vdots \\ \perp \end{smallmatrix}}}{\neg \phi} \neg i \quad \frac{\phi \quad \neg \phi}{\perp} \neg e$$

$$\perp \quad \text{(no introduction rule for } \perp \text{)} \quad \frac{\perp}{\phi} \perp e$$

$$\neg \neg \quad \frac{\neg \neg \phi}{\phi} \neg \neg e$$

Some useful derived rules:

$$\frac{\phi \rightarrow \psi \quad \neg \psi}{\neg \phi} \text{MT} \quad \frac{\phi}{\neg \neg \phi} \neg \neg i$$

$$\frac{\boxed{\begin{smallmatrix} \neg \phi \\ \vdots \\ \perp \end{smallmatrix}}}{\phi} \text{PBC}$$

$$\frac{}{\phi \vee \neg \phi} \text{LEM}$$

Dodatak: bitna pravila prirodnog zaključivanja u propozicijskoj logici

- Izvor: Huth, Ryan, **Logic in Computer Science**, Cambridge University Press, 2004.
- Objašnjenja nekih pojmova:
 $\rightarrow e$ = modus ponens
 MT = modus tollens
 PBC (Proof By Contradiction)
 LEM (Law of the Excluded Middle) = TND (Tertium Non Datur) = Trećega nema